Statistical Learning (MT7038) - Project 2

Instructions: *This project consists of 3 tasks that should be solved individually. Unless it is specified in the task, you are free to use any R-package in this project.*

The solution should be submitted at the course webpage in a single .pdf file with your source code attached as appendices. Your source code should include clear comments and documentations to describe what are evaluating.

TASK 1 (k-mean)

This task allows you to experience the performance of k-mean clustering and its possible limitations when applying to imbalance data and non-spherical symmetric data. The data for this task are from the text files “Imbalance_Data.txt” and “Star_Data.txt”. The first and second columns of the data files are the x_1 and x_2 coordinates of the 2D feature vector, respectively.

a) For the “Star_Data.txt” data, write a simple code (you can use the `kmeans()` function in R) to cluster the data with number of clusters $N_c = 2$, 3, and 4. Plot the clustering results on the x_1 and x_2 plane for $N_c = 2$, 3, and 4.

b) Discuss the performance of the k-mean clustering from the plots in a). If the clustering results do not look good, suggest a reasonable way for improvement and justify your answer.

c) Repeat part a) and b) above using the data file “Imbalance_Data.txt”.

TASK 2 (PCA)

In this task, you will implement PCA as an eigen-problem and understand its difference from linear regression, besides one is unsupervised and one is supervised. This task uses the data from the file “PCA_Data.txt”. The first and second columns of the data file are the x_1 and x_2 coordinates of the 2D feature vector, respectively.

a) Write a simple code to compute the 2 PCs from the data points. The code should perform a diagonalization of the corresponding covariance matrix to find the eigenvectors. The built-in R functions `prcomp()` and `princomp()` should not be used in this exercise. Plot the 2 PCs together with the data points. Note: Make sure to center the data points correctly such that the PCs should go through the mean location of the data points.

b) Now treat the x_2 and x_1 coordinates of the data as the response and predictor variables, respectively, and perform a linear regression for the data using the least square fitting (i.e., a supervised learning). Plot the linear regression line together with the first PC and the data points. Is the linear regression line the same as the first PC?

TASK 3 (SVM)

This task allows you to experience the usage of `svm()` function in R and explore a little bit how to generalize to nonlinear decision boundary. This task uses the "mixture simulation" data in the course book. To download the data, go to the webpage of the course book (https://web.stanford.edu/~hastie/ElemStatLearn/), and then click "Data" on the left panel to find the data and follow the instruction to use them.

a) As a warm up exercise, perform the support vector machine for linear classification for the data using the `svm()` function. To use the `svm()` function, you need to first install the R-package e1071 (run `install.package("e1071")` to install). Reproduce the two figures in Fig. 12.2 in the course book with $C = 10000$ and $C = 0.01$. Hint: use the C-classification and the linear kernel in the `svm()` function.

b) At the time when you are working on this task, we have not yet discussed the "kernel trick" in the lecture to extend support vector machine to nonlinear decision boundaries. Nevertheless, we can still perform a "simpler" version of the nonlinear extension by considering an enlarged feature space with 5 coordinates $x_1, x_2, x_1x_2, x_1^2, x_2^2$, where x_1, x_2 are the original features of the data. Perform the same procedure of linear support vector classification (with $C = 10000$ and $C = 0.01$) as in part a) for the data in the enlarged feature space. Then plot the new nonlinear decision boundaries and their margins (i.e., those correspond to the dark dash lines in Fig. 12.2) together with the data points in the original x_1 vs x_2 plane. Do the new nonlinear decision boundaries look reasonable?